GRIDCOLE: a Grid Collaborative Learning Environment

Miguel L. Bote-Lorenzo, Luis M. Vaquero-González, Guillermo Vega-Gorgojo, Juan I. Asensio-Pérez, Eduardo Gómez-Sánchez, Yannis A. Dimitriadis
School of Telecommunications Engineering, University of Valladolid
Camino Viejo del Cementerio s/n, 47011 Valladolid, Spain
{migbot@, lvaqgon@ulises, guiveg@, juaase@, edugom@, yannis@}tel.uva.es

Abstract

Research in the Computer Supported Collaborative Learning (CSCL) field is currently tackling three important problems which are closely related: difficult integration of CSCL tools, scarce software reuse and technification. This paper presents a Grid Collaborative Learning Environment (GRIDCOLE) that combines IMS Learning Design (IMS-LD) and Open Grid Services Architecture (OGSA) technologies in order to address these issues. More specifically, GRIDCOLE allows users with no technical skills easy integration and use of applications that effectively support collaborative learning processes. IMS-LD is employed to formally describe the design of these collaborative learning processes while applications are built using tools provided by third-party suppliers as OGSA-compliant grid services. With this aim, GRIDCOLE architecture and its main functionalities are introduced. A simple prototype is also presented in order to show the feasibility of the most important concepts introduced in this paper. Furthermore, a real collaborative learning scenario in which we plan to validate GRIDCOLE is introduced.

1. Introduction

Computer Supported Collaborative Learning (CSCL) is a discipline devoted to research in educational technologies that focuses on the use of Information and Communications Technology (ICT) as a mediational tool within collaborative methods of learning [6,23]. Examples of these methods include peer learning and tutoring, reciprocal teaching, project or problem-based learning, simulations and games.

Although CSCL is already a mature research field, there is still a number of difficult challenges currently being studied. Among these problems there are three which are closely related. First, CSCL tools are difficult to integrate inside larger applications due to their lack of clear standardized interfaces and incompatibility between conceptual models and design foundations [4]. Besides, scarce reuse of educational software owes to the use of application building blocks representing low level abstractions that do not fit with the mental model of educators, which often use these blocks to build customized applications [19]. Finally, the technification problem refers to the need of technical skills that makes it difficult for teachers and students to use learning systems [21]. Significantly, two different standards have appeared recently that can help to address these issues.

On the one hand, the Open Grid Services Architecture (OGSA) [9] has emerged as a de facto standard for the construction of grid systems [3,8]. Building on concepts from both the grid and web services communities, OGSA defines a set of conventions and uniform semantics that support dynamic integration of grid services within larger applications. The CSCL domain could benefit from this property by adhering to the OGSA standard to integrate CSCL tools offered as grid services. Such approach would also promote CSCL tool reuse, since grid services are application building blocks that would represent high-level abstractions closer to educators’ mental model. The fact that OGSA philosophy pushes deployment set-up responsibilities to the grid service provider would reduce the technification problem too. Moreover, the use of an OGSA-based grid infrastructure for the integration of CSCL tools could bring additional advantages. These include new learning scenarios [11] or enabling easy integration of tools requiring supercomputational capabilities within CSCL applications similar to CoVis [17], NICE [20] and COVASE [16].

On the other hand, the IMS Learning Design (IMS-LD) specification [14] provides a framework that enables formally describing the design of teaching-learning processes for a wide range of pedagogies in online learning. More specifically, collaborative learning designs can be expressed using this specification [5,13]. Again, the use of IMS-LD standards may be beneficial for the CSCL domain. In this sense, it is noteworthy that IMS-LD follows the Activity Theory (AT) [5], which provides a suitable conceptual model so as to enable easy CSCL tool integration [4]. IMS-LD also promotes the reuse of learning designs and, as a consequence, the reuse of software tools that may be employed to support them. The fact that no technical knowledge or skills are required to use IMS-LD facilitates the labor of educators when...
defining a learning design. Furthermore, IMS-LD learning designs can be interpreted using workflow technology thus enabling learning activity management so as to assist students in achieving desired learning objectives [5,22].

Within this framework, in this paper we present a Grid Collaborative Learning Environment (GRIDCOLE) that pretends to leverage the advantages of both IMS-LD and OGSA-based grid technologies for the CSCL domain. GRIDCOLE allows educators easy tool integration within a single application aiming at supporting a given collaborative learning design defined with IMS-LD. These tools will be supplied by third-party service providers as OGSA compliant grid services. Furthermore, applications within GRIDCOLE will provide proper activity management support.

The rest of this paper is organized as follows. Section 2 discusses how both IMS-LD and OGSA technologies can be combined in order to provide effective support for the realization of collaborative learning activities while addressing the difficult task of tool integration and scarce software reuse. Section 3 presents an overview of GRIDCOLE architecture and its main functionalities showing that suitable applications for collaborative learning can be integrated and employed by users with no technical skills thus avoiding the technification problem. Section 4 introduces a simple GRIDCOLE prototype that has already been developed in order to prove the feasibility of this approach. In addition, section 5 describes a real collaborative learning scenario in which we plan to validate GRIDCOLE. Section 6 presents some works related to grid and collaborative learning topics. Finally, conclusions and future work can be found in section 7.

2. IMS-LD and OGSA grids for collaborative learning

GRIDCOLE pretends to leverage the advantages of both IMS-LD and OGSA grids for the CSCL domain. This section shows how a collaborative learning design described using IMS-LD can be supported using OGSA-compliant grid services in order to achieve such a goal.

The IMS-LD specification defines a structured XML-based language that can be employed to formally express learning designs. A learning design is a description of a method enabling learners to attain desired learning objectives by performing predefined learning activities in a certain order within the context of a given learning environment [14].

More specifically, IMS-LD allows to formally describe a learning design in terms of a learning workflow, and a set of environments. The learning workflow specifies the sequence of activities that learners should perform in order to reach predefined learning objectives according to the different roles that they may play in a learning design. The environments are defined by the resources, i.e. the tools and contents, that should assist learners during the realization of each activity according to the role played. IMS-LD can also be employed to describe collaborative learning designs that usually involve the realization of both collaborative and non collaborative activities.

Three different elements can thus be employed to support the realization of a collaborative learning design: a Learning Flow Engine (LFE), tools and content documents. A LFE interprets the learning workflow described in the design and coordinates the executions of multiple activities instructing who (users) does what (activities), using which (tools and documents) and when. In other words, an LFE is responsible for automatically scheduling the activities to be performed by each user as well as providing suitable tools and contents in order to properly assist him during the realization of each activity.

A generic LFE for IMS-LD designs can be implemented as an OGSA-compliant grid service. This way, the same LFE service (though different instances) can be employed to support different collaborative learning designs, thus promoting its reuse.

The tools used to assist learners can also be supplied by service providers adhering to OGSA standards. In this sense, tools could be either collaborative (e.g. concept map generation tools, editors, electronic whiteboards, games) or non-collaborative (e.g. document search tools and simulators). Users can thus cooperate employing the same instance of a CSCL tool. On the contrary, different instances of the same service should be created for the users of a non-collaborative tool. Again, the combination of IMS-LD and OGSA enables reuse of tool grid services to support different collaborative learning designs.

Finally, content documents required during the realization of a collaborative learning design can be supplied by third-party providers. In this case, documents are URL-addressable resources that may be (or not) hosted by an OGSA-compliant grid service.

It is noteworthy that the joint usage of OGSA and IMS-LD standards in order to support collaborative learning designs leads to coarse grained blocks: LFE grid service, tool grid services and URL-addressable content documents. These blocks represent high level abstractions understandable by educators. Therefore, they can be employed as building blocks for applications designed by educators aiming at supporting collaborative learning designs.

3. GRIDCOLE overview

GRIDCOLE is a Grid Computing Environment (GCE) [10] that allows users with no technical skills easy access
to grid resources and applications in order to provide effective support for collaborative learning. More specifically, GRIDCOLE allows educators easy integration of applications that can support a collaborative learning design using the building blocks described in previous section. Furthermore, it allows learners easy participation in collaborative learning designs using these applications. This section provides an overview on how this is achieved in GRIDCOLE environment.

With this aim, the design and implementation goals of GRIDCOLE environment are first stated. Next, its architecture is presented and discussed. Following, it is shown how users with no technical skills are assisted in GRIDCOLE. Finally, the low-level process of supporting a collaborative learning design in GRIDCOLE is outlined.

3.1. Design and implementation goals

GRIDCOLE environment has been designed and is currently being implemented pursuing the following goals:

- **IMS-LD support.** The IMS-LD specification allows the definition of rich collaborative learning designs in a standard way. Applications integrated within GRIDCOLE should be able to properly support IMS-LD based collaborative learning designs.

- **OGSA compliance.** The service-oriented view promoted by OGSA enables easy integration of coarse grained grid services within a single application. Applications integrated within GRIDCOLE should be built using grid services adhering to OGSA specifications.

- **Easy application integration.** Grid service integration may be a daunting task for an educator with little or no technical skills. GRIDCOLE should provide suitable facilities so as to enable educators easy integration of GRIDCOLE applications.

- **Easy application set-up.** A big impediment to the acceptance of software by user communities, and specially in education, is the complexity of the compilation, installation, and configuration of that software. Whenever they cannot be avoided, these tasks should be carried out in GRIDCOLE with minimum user intervention.

- **Low requirements on client machines.** Placing software requirements on client machines which are hard to meet would discourage user participation. Minimum requirements on client machines should thus be kept as low as possible.

- **Low requirements on service providers.** Grid services are usually supplied by third-party service providers. This promotes the existence of a wide range of services that can be chosen for application integration. However, service providers are not likely to adhere to non widely accepted standards. Thus, minimum requirements should be placed on providers in order not to dissuade them from offering services that can be integrated in GRIDCOLE.

3.2. GRIDCOLE architecture

The general GRIDCOLE architecture can be decomposed into three main elements: Web Portal, Applications, and Application Clients. The breakdown of the architecture is shown in Fig. 1.

**Figure 1. GRIDCOLE three-tier architecture**

As it is usual in many other grid computing environments [10], a **Web Portal** provides a single access point to GRIDCOLE using a simple web browser. Moreover, the portal is responsible for assisting authenticated users when operating within the environment. In order to achieve such a goal, the portal counts on a number of elements.

First, a **Database** keeps administrative information about users such as logins, passwords, and user profiles. This database stores information about applications currently running within the environment as well as the list of users which are allowed to use them. Furthermore, a **Repository** is employed to store collaborative learning designs as well as information about applications that have been already integrated. Finally, a **Tool Searcher** allows educators easy search for tool services to be integrated in an application on a keyword description basis. The tool searcher uses this description to query the index services where service providers register their tool services and then returns a list of those grid services that best match the educator’s request as well as their location.
A set of a Learning Flow Engine grid service, some Tool Grid Services, and URL-addressable Content Documents that support a given collaborative learning design is called a GRIDCOLE Application. These applications can be integrated and launched by educators using the Web Portal. Applications are thus inherently distributed, since the tools employed are executed in computational resources owned by service providers, allowing for the integration of tools requiring supercomputational capabilities.

A GRIDCOLE Application Client can be downloaded and installed from the web portal in order to use GRIDCOLE applications. The application client provides a desktop-like graphical user interface (GUI) (called GRIDCOLE Desktop) that allows users easy collaboration and interaction with tool services and content documents. With this aim, the application client automatically downloads necessary service clients and document browsers.

A Service Client is a software component providing a graphical interface that allows user easy interaction with a specific tool service. Service clients are supplied by service providers and include all necessary stubs to communicate with the corresponding grid service. On the contrary, Document Browsers are supplied by the web portal. Both service clients and document browsers provide GUIs which are properly integrated within GRIDCOLE desktop in the form of panes.

The need of an application client in order to use a GRIDCOLE application imposes an extra requirement on the client machine where it must be installed. Nevertheless, such a standalone client is necessary in order to enable highly interactive interfaces that cannot be provided using web technologies such as applets, servlets or portlets. Moreover, if all users interacted with content documents and tool services through the portal, a communication bottleneck would arise soon. Minimum requirements on the client machine can be kept reasonably low if the application client is developed using Java. Thus, the user only needs to install a Java Virtual Machine and the application client in order to use any application.

An extra requirement is also placed on third-party service providers which are asked to provide a service client, because only the service provider can determine a proper graphical interface layout and suitable logic for a service client that allows an easy interaction of the user with the tool services provided. Again, in order to keep requirements as low as possible, service clients can be delivered in the form of widely-accepted Java Beans [18].

3.3. GRIDCOLE functionality

GRIDCOLE environment offers different facilities according to the three different types of user profile. Educators are allowed to store collaborative learning designs. These designs can be later retrieved by educators to integrate GRIDCOLE applications. Besides, educators can launch or stop these applications. Both educators and learners are enabled to participate in a the execution of collaborative learning designs. Finally, administrators carry out several tasks related to user management such as creation or deletion of user accounts.

Educators and learners are users which are not expected to have any technical skills. This subsection shows how the web portal assist these users in order to perform the following operations: application integration, application launching and joining the execution of a collaborative learning design.

3.3.1. Integration of GRIDCOLE applications. A collaborative learning design uses IMS-LD to specify the tools and contents to be used in its execution. This specification is abstract, i.e. a collaborative learning design describes what tools and contents should be employed, but not where these resources can be found within the runtime environment. Thus, application integration in GRIDCOLE is as easy as providing the location of the resources to be used within the OGSA grid environment in which GRIDCOLE is deployed.

The web portal assists and guides the user all over the application integration process. First, the educator must select a collaborative learning design which was previously stored in the repository. Then, for each tool that is declared in the chosen collaborative learning design, the portal asks the educator for the location of a factory that can create such a service at runtime. The educator can use the tool searcher facility in order to find suitable tools for the application being integrated. Correspondingly, the educator is asked for an URL for each content document declared in the collaborative learning design.

Once the locations of all resources have been provided, the application is integrated. Next, the information of the application, i.e. the collaborative learning design description and the locations of the resources to be used, is packaged into a single file following the IMS Content Packaging [15] specification, which has been slightly extended in order to support the addressing of resources within an OGSA environment. The resulting package is called following IMS terminology, a collaborative learning unit. Collaborative learning units can be stored in the repository and later retrieved by educators in order to launch a GRIDCOLE application.

3.3.2. Launching a GRIDCOLE application. The execution of a collaborative learning design starts when the educator launches a GRIDCOLE application. Again, the web portal assists educators to perform this task in
two easy steps. First, the educator must provide a list of participants for the run. Participants are chosen from the users list stored in database. Then, the educator selects the collaborative learning unit, i.e. the application, to be launched from the repository.

As a result of the launching operation, a new Learning Flow Engine instance is created. Then, the LFE is registered in the database and provided with all information necessary to coordinate the execution of the collaborative learning design. Once this is done, the application is ready to be used.

3.3.3. Joining a collaborative learning design.
Educators and learners willing to join the execution of a collaborative learning design use the web portal to query the database. This way they can find out all the applications currently running that they are allowed to use. Once the user finds the desired application, the portal automatically notifies to the user’s application client the location of the LFE corresponding to that application. The user can then participate in the GRIDCOLE application.

3.4. GRIDCOLE application execution

Once the application client knows the location (i.e. the GSH) of a LFE, a sequence of low-level activities are performed by both application client and the LFE in order to allow application execution. These activities are shown in Fig. 2 and described next.

**Figure 2. Activities performed by application clients and LFE**

First, the application client registers in the LFE. Registration is only possible if the user is in the list of allowed participants. If registration succeeds, the LFE provides the application clients with a list of URLs pointing at the service clients and document browsers required for current activity. While the application client is downloading service clients, the LFE creates instances of the tool services scheduled to support this learning activity, using the GSHs of the corresponding grid service factories. If the tool is collaborative, one instance of the corresponding service is created and shared by each group of collaborators. Otherwise, different instances are created for each user.

Next, the LFE provides the application clients with a list of GSHs and URLs pointing to the tool service instances and content documents supplied in order to assist the user while carrying out the scheduled learning activity. Then, the application client instantiates the downloaded service clients and document browsers and provides them with the corresponding GSHs and URLs.

Users can then perform the learning activity assisted by scheduled documents and tools. The application client presents all service clients and document browsers as graphical interface panes integrated within a single user desktop. Users can thus employ panes to fruitfully collaborate with other participants by means of collaborative tools, to work with non-collaborative tools, or to browse content documents.

The LFE may assign a new activity to a user when he has completed the current one. In this case, the LFE notifies the application client that current activity is over. As a result, the application client disables the use of current tools and document browser. Service clients and document browsers which are going to be used in the following activities are kept in the application clients, while the rest are removed. Then, the process of loading new service clients and document browsers and binding them to the corresponding tool services is carried out for the new activity.

4. GRIDCOLE prototype

GRIDCOLE environment is currently under development using Globus Toolkit 3 (GT3) [1]. However, a first prototype has already been built in order to prove the feasibility of some of the concepts presented in this paper.

More specifically, a first version of the tool searcher and the application client have been developed and integrated in a simple web portal. The main limitation of this first version of the application client is that it requires an installation of the GT3 core to work.

A limited Web Portal assistance facility for application launching has also been developed. However, assistance facilities for application integration, launching and joining a collaborative learning design are still under way, and these tasks are currently hand made.

Furthermore, a simple GRIDCOLE application has been developed. This application supports an extremely simple collaborative learning design described using a simplification of IMS-LD language. The collaborative
learning design consists of three activities that must be performed by four users with limited time. In the first activity, users share a chat tool in groups of two that they can use to debate. In the second activity, each user has an individual counter tool. In the third activity, a chat tool is shared by all four users and, again, a counter tool is provided to all users.

A limited LFE grid service has been developed in order to interpret the aforementioned simplified IMS-LD language. Both chat and counter grid services have also been built as well as their corresponding service clients.

5. Sample GRIDCOLE application

Once the development of GRIDCOLE is completed, a real application will be evaluated in a real collaborative learning scenario. In order to better illustrate the potential benefits of GRIDCOLE, this section first describes the context and learning objectives of such a collaborative learning scenario and then outlines a GRIDCOLE application that could effectively support it.

5.1. Application context and learning objectives

This application is to be used within a course on Computer Architecture for undergraduate students at the University of Valladolid, Spain. This course is organized around a design project which in turn is divided into three subprojects (see [7] for details). In this project, students organize in groups of four and collaboratively play the role of consultants that have to advise on a computing solution (machine, operating system, software, network, etc.) for a given customer, which is played by the teacher.

The application described here applies for the first subproject, in which students get to know the client, model the customer’s presumed computational load by mixing standard benchmarks, test real machines using the benchmarks, and finally make a recommendation to their client. The subproject pursues clear learning objectives. On the content side, it is expected that students learn how to use benchmarks, and get a quantitative impression on a few real machines (with different CPUs, memories, etc.). Moreover, the subproject also aims at promoting several abilities, such as interpreting and selecting information (to find out more about the customer, to understand the benchmark results), arguing and taking a compromise solution (to model the computational load, or to make the final suggestion). The subproject lasts for six two-hour sessions on approximately four weeks.

5.2. Application outline

The sequence of activities that will be managed by the application LFE is shown in Fig. 4. These activities, as well as the tools and documents that the GRIDCOLE application will provide in order to support them are described next.

![Figure 4. Sequence of activities to be managed by the GRIDCOLE application's LFE.](image-url)
manages and supports the aforementioned learning activities. With this purpose, the educator will be able to select a number of collaborative and non-collaborative tools such as:

- A group formation tool that may assist students to select their partners when organizing in groups.
- A collaborative concept map tool that may aid students to understand client needs.
- A collaborative questionnaire tool in order to allow students to fill required forms.
- A voting tool so as to assist decision-making processes.
- A task assignment tool that may help students determine who is in charge of benchmarking which machines.
- Benchmarking tools that execute weighted benchmarks on remote machines and collect the results.
- A text editor to allow students to write down significant results and comments.
- A collaborative debate tool in order to enable and promote discussions between students.
- A collaborative editor tool that may allow students to collaboratively generate text documents.

It is noteworthy that all tools described here are highly reusable in different learning scenarios and that this reuse possibility would be further promoted by the use both IMS-LD and OGSA standards within GRIDCOLE environment. Moreover, it can be appreciated the use of an OGSA grid provides an additional important advantage for this collaborative learning scenario since it would allow the execution of benchmarks in a wide range of real machines owed by third-party service providers.

6. Related work

Several works can be found in literature related to the use of grid computing for collaborative learning. For example, GIMOLOUS [24], is an e-learning system based on an OGSA grid for education in the environmental science domain. Nevertheless, it does not provide support for collaborative activities neither for activity management. ISIlab [2] is a remote laboratory for teaching electronics also based in a grid infrastructure. Again, collaborative learning and activity management are not supported. DARE [4] is a collaborative learning environment that introduces an Activity Theory-based conceptual model to enable CSCL tool integration thus similar to the conceptual model employed in GRIDCOLE. However, DARE does not hold the activity management facilities derived from the use of IMS-LD nor the advantages enabled by OGSA grids. COW [22] is a learning flow engine for the interpretation of IMS-LD designs using XPLD language. Some of the ideas underlying this engine could be reused for the development of GRIDCOLE’s LFE.

It is also worth mentioning the existence of web services [12] technology to which grid services are closely related. Web services may also be employed as application building blocks representing high-level abstractions. However, the use of grid services provides a number of extra values to GRIDCOLE when compared to web services. Examples include allowing sharing of scarce resources, and enabling use of supercomputing capabilities. Moreover, grid service technology offers extra features which are required to develop collaborative tools such as notification, state, and lifetime management.

7. Conclusions and future work

Research in the CSCL field is currently tackling three important problems which are closely related: difficult integration of CSCL tools, scarce software reuse and technification. This paper has presented GRIDCOLE, a Grid Collaborative Learning Environment that addresses these issues by combining OGSA and IMS-LD standards. It has been shown that both standards may be employed to effectively support collaborative learning while promoting software reuse and enabling easy tool integration. Building on these ideas, GRIDCOLE architecture and its main functionalities were introduced and discussed showing that users with no technical skills can use and integrate GRIDCOLE applications. Such applications consist of a elements consisting in Learning Flow Engine, tool grid services and content documents aiming at effectively supporting a collaborative learning design, that is formally defined using IMS-LD. GRIDCOLE is currently under development, but a prototype has been presented showing the feasibility of the main underlying ideas. Moreover, a real collaborative learning scenario in which we plan to validate GRIDCOLE was introduced and discussed in order to better illustrate its potential benefits.

However, much work has still to be done. In addition to the complete development and evaluation of GRIDCOLE, future work includes the addition of awareness facilities to GRIDCOLE applications. A framework of useful collaborative and non-collaborative grid services will also be researched. Efforts will be made in order to characterize these services using the Learning Object Metadata (LOM) standard. This would enable the possibility of further automating the instantiation process for tool services. Moreover, it would be convenient to suppress the need of service providers to offer the corresponding service clients for download, or at least to find simpler ways to offer them. The User Interface Markup Language (UIML) specification will be studied for this purpose. In addition, if IMS-LD is found to be
very rigid to organize collaborative activities, the use of Peer to Peer (P2P) tools as a complementary channel for collaboration may alleviate this. Finally, scheduling facilities which are usually employed in grid computing will be studied in order to provide collaborative tools according to predefined Quality of Service (QoS) parameters.

Acknowledgements

This work has been partially funded by Spanish Ministry of Science and Technology projects TIC2002-04258-C03-02, TIC2000-1054 and Castilla y León Regional Government project VA 117/01. The authors would also like to thank the “Intelligent & Cooperative Systems Research Group” at the University of Valladolid, and specially Pablo Orozco, Davinia Hernández, Bartolomé Rubia, Iván Jorrín, and Alejandra Martínez, for their support and ideas.

References